Academic Search Premier; Computer Science Index; Computer Source; Internet and Personal Computing Abstracts; Information Science & Technology Abstracts for *error W2 inject** Add this search to

Searched: folder | Display link to this search

Database Help

Find: error W2 inject* Search Clear

Search Tips

Folder is empty.

✓ Limiters set
Refine Search | Results

To store items added to the folder for a future session, Sign In to My EBSCOhost

1 - 3 of 3 Pages: 1

Sort by : Date

Add (1-3)

The number of available results reflects the removal of duplicates.

1. Design and evaluation of system-level checks for on-line control flow error detection. By: Alkhalifa, Z.; Nair, V.S.S.. IEEE Transactions on Parallel & Distributed Systems, Jun99, Vol. 10 Issue 6, p627, 15p, 7 charts, 9 diagrams, 4bw; Abstract: Evaluates the fault detection capabilities of the control-flow checking systems installed in real-time computer applications using fault and error injection. Enhanced control-flow checking using assertions (ECCA); Theoretical determination of the capabilities of ECCA; Table of selected transient error models.; (*AN 2177048*)

Add

2. FINE: A Fault Injection and Monitoring Environment for Tracing the UNIX System Behavior under Faults. By: Wei-lun Kao; Iyer, Ravishankar K.; Tang, Dong. IEEE Transactions on Software Engineering, Nov93, Vol. 19 Issue 11, p1105, 14p, 11 charts, 4 diagrams, 6 graphs; Abstract: Fault injection has been used to evaluate the dependability of computer systems, but most fault-injection studies concentrate on the final impact of faults on the system with an emphasis on fault latency and coverage issues. What realty happens after a fault is injected and how a fault propagates in a software system are not well understood. This paper presents a fault injection and monitoring environment (FINE) as a tool to study fault propagation in the UNIX kernel. FINE injects hardware-induced software errors and software faults into the UNIX kernel and traces the execution flow and key variables of the kernel. It consists of a fault injector, a software monitor, a workload generator, a controller, and several analysis utilities. Experiments on SunOS 4.1.2 are conducted by applying FINE to investigate fault propagation and to evaluate the impact of various types of faults. Fault propagation models are built for both hardware and software faults. Transient Markov reward analysis is performed based on the models to evaluate the loss of performance due to an injected fault. Experimental results show that memory faults and software faults usually have a very long latency while bus faults and CPU faults tend to crash the system immediately. About half of the detected errors are data faults, which are detected when the system tries to access an unauthorized memory location. Only about 8% of faults propagate to other UNIX subsystems. Markov reward analysis shows that the performance loss incurred by bus faults and CPU faults is much higher than that incurred by software and memory faults. Among software faults, the impact of pointer faults is higher than that of nonpointer faults. [ABSTRACT FROM AUTHOR]; (*AN 14325387*)

Add

3. Fault Injection for Dependability Validation: A Methodology and Some Applications. By: Arlat, Jean; Aguera, Martine; Amat, Louis; Crouzet, Yves; Fabre, Jean-Charles; Laprie, Jean-Claude; Martins, Eliane; Powell, David. IEEE Transactions on Software Engineering, Feb90, Vol. 16 Issue 2, p166, 17p, 13 diagrams, 4 graphs, 8c; Abstract: This paper addresses the problem of the dependability validation of fault-tolerant computing systems and more specifically the validation of the fault-tolerance mechanisms. The presented approach is based on the use of fault-injection at the physical level on a hardware/software prototype of the considered system. The place of this approach in a validation directed design process, as well as its place with respect to related works on fault-injection, is clearly identified. The major requirements and problems related to the development and application of a

Add

OPTION 1
Enter keywords or phrases, select fields, and select operators

in All Fields

AND in All Fields

AND in All Fields

Run Search Reset

OPTION 2
Enter keywords, phrases, or a Boolean expression

```
error <near/2> inject*
```

Run Search Reset

» Note: You may use the search operators <and> or <or> without the start and end brackets <>.

» Learn more about Field Codes, Search Examples, and Search Operators

» Publications

- Select publications
  - IEEE Periodicals
  - IEE Periodicals
  - IEEE Conference Proceedings
  - IEE Conference Proceedings
  - IEEE Standards

» Select date range

- Search latest content update (23 May 2005)
- From year All to 2001

» Display Format

- Citation
- Citation & Abstract

» Organize results

Maximum 100

Display 100 results per page

Sort by Relevance

In Descending order